A Case-study based on French Resources for the Semantic Analysis of Sentences in Coq

Line Jakubiec

A Case-study based on French Resources for the Semantic Analysis of Sentences in Coq

Line Jakubiec-Jamet
LIF-CNRS UMR 7279, Aix Marseille University
Parc Scientifique et Technologique de Luminy
163 avenue de Luminy - Case 901, F-13288 Marseille Cedex 9, France
Line.Jakubiec@lif.univ-mrs.fr

Abstract

This paper presents an approach devoted to the formalization of sentence frames in the Coq system. Then, we instanciate these frames for performing a semantic analysis of simple sentences. We rely on a hierarchy of types for type-checking the conceptual well-formedness of sentences. To do so, we investigate how to exploit the particular features of the Coq type system for encoding the syntax-semantics interface and then we show how to combine it with large resources for french language : our hierarchy of type is based on the freely available french wordnet named WOLF; our lexicon can be easily extended using the dictionary of french verbs developed by Dubois and Dubois-Charlier.

1 Introduction

During the last fifteen years, the use of type theoretic methods for describing natural langage syntax and semantics has gained more and more popularity (RETORÉ, 2001) (LECOMTE, 1996) (DE GROOTE and RÉTORÉ, 1996), resulting in the development of software relying on these methods. First, these type theoretic methods can be constantly improved with advances in the field of logic. Secondly, it is important to evaluate them on real linguistic resources. This paper relies on WOLF (HANOKA and SAGOT, 2012), a french wordnet (FELLBAUM, 1998) that is used to represent our hierarchy of types and on the LVF dictionary (DUBOIS and DUBOIS-CHARLIER, 1997) that is used as the lexicon of verbs. This work focusses on a Coq formalization (Coq, 2014) of the syntax-semantics interface and it proposes an integration of lexical resources into the Coq logic-based system.

2 Related work

Among the theoretical achievements, let us mention the works on categorial grammars which provide the integration of syntax and semantics in the same framework as it is described in (MOORTGAT, 1996). Categorial grammars are lexicalized that means that all items in the lexicon are typed. Although they describe syntactical rules, they also preserve the compositional aspect of Montague semantics. Experiences have been performed in (MOORTGAT and MOOT, 2002); it shows computational facilities of a logic tool based on categorial grammars using a Dutch lexicon. The most well-known categorial grammars are those based on Lambek-Calculus. Although many studies have already showed that it is natural to associate a syntactic term to a semantic type and the reverse (RANTA, 2009) (MOOT and RÉTORÉ, 2011) (MUSKENS, 2011), our approach implemented in Coq focusses on the generality of definitions leading to reusable specifications. In Coq, few investigations have been performed for natural language processing. An algorithm is developed in (COSCOY, 2000) ; it produces natural language sentences from proofs described in a mathematical language. Later, for the case of categorial grammars, (ANOUN, 2005) gave a Coq formalization of the Lambek-Calculus and of an extension as multimodal grammars. In (LUO, 2010) and (LUO, 2012), the author studies specific features of type theories to obtain more expressivity in formal semantics.

3 The Approach using Lexical Ressources

3.1 Ontology as Concept Hierarchy, based on WOLF

For the paper, the verification of the sentence's conceptual well-formedness will be illustrated by a small part of a conceptual tree built from WOLF.

The path of the tree will be the following:

```
transport -> vehicule ->
             vehiculeAmoteur -> voiture
```

It describes a world from which it is possible to make sentences. This world is organized from $transport$. For example, a $vehicule$ is classified into the $transport$ category. In this path, each node is labelled by a conceptual information that can be specified as a type. Thus, for organizing this information, it is natural to consider the subtyping principle. In typed definitions, a subtype may appear wherever an element of the super type is expected. More generally, in our verb's semantic representation, a type t_1 is compatible with a type t, if t_1 is a subtype of t. Consequently, a semantic representation parameterized with t will be valid for parameters of type t_1 and for all those of the lower part.

In Coq, we declare the conceptual types ($transport$, $vehicule$...) as logical propositions. Then, we use the coercion mechanism for describing the relations between types, as follows :

```
Coercion vehicule_is_transport :
  vehicule >-> transport.

Coercion vehiculeAmoteur_is_vehicule :
  vehiculeAmoteur >-> vehicule.

Coercion voiture_is_vehiculeAmoteur :
  voiture >-> vehiculeAmoteur.
```

Each coercion creates a path between two nodes of the tree. The whole list of coercions is ordered and it defines a conceptual hierarchy. Coq detects ambiguous paths during the creation of the whole tree (for example a tree that describes all the means of transport) and, it verifies the *uniform inheritance condition* where at most one path must be declared between two nodes.

3.2 Coq Semantic Representation of Sentences

3.2.1 Lexicon of Verbs, based on the LVF dictionary

For typing the verb's representation, we just use the conceptual types described in the section 3.1.

For each verb, we have to choose the best label which must correspond to the smaller type in the hierarchy. The LVF dictionary contains an exhaustive description of verbs with their formal properties and their semantic classification. Moreover, their scope is defined by syntax. This allows to describe the use cases of verbs (a verb can be used only with a subject or with a subject and a complement and so on). For our study, we rely on a class dedicated to the $movement$ verbs.

For example, the verb $to\ move$ ($rouler$ in LVF that is defined on the domain VEH) can be used for all the vehicules. So, it is declared in Coq as a logical proposition by the unary predicate $move$ as follows :

```
Parameter move : vehicule -> Prop.
```

3.2.2 Sentences as logical expressions

Let us consider the sentence *A car is moving*. It can be represented by the logical expression : $\exists x, move(x) \wedge is\_car(x)$ where is_car is a unary predicate of type $vehiculeAmoteur \rightarrow Prop$. Finally, the sentence can be represented in Coq as follows :

```
Definition a_car_is_moving :=
  exists x, move(x) /\ is_car(x).
```

But this specification is clumsy because it does not use the expressiveness of the Coq language.

3.2.3 Towards Generic Models

The category of sentences that we study is composed of a verb and a subject. Due to the expressiveness of the Coq language, we can generalize the representation given in the previous paragraph (in order to reuse it later), by specifying a general frame that states : $\exists x, verb0(x) \wedge is\_something(x)$, where $verb0$ and $is\_something$ are polymorphic predicates respectively parameterized on $A1$ and A, with $A1$ subtype of A. The complete specification in Coq is given below, inside a section :

```
Section General_frame_v0.

(** Local parameters of the section **)
  Variables (A A1:Prop)
            (A1A : A1 -> A).

(** Declaration of the subtype **)
  Coercion A1A : A1 >-> A.

(** Declaration of the predicates **)
  Variables (verb0 : A1 -> Prop)
            (is_something : A -> Prop).

 (** Definition of the generic model **)
  Definition frame_verb0 := exists c,
   verb0(c) /\ is_something(c).

End General_frame_v0.
```

Technically, in the definition, the variable c is of type $A1$ ($A1$ is an implicit type). Outside the

section, the local context of the definition is discharged. This means that A, $A1$, $A1A$, $verb0$ and $is\_something$ will appear as parameters of the definition $frame\_verb0$. So, $frame\_verb0$ depends on two types, on a coercion which states a subtyping relation and on two predicates. It is a generic representation thanks to polymorphism (from the parameters A and $A1$) and higher-order (from the $verb0$ and $is\_something$ predicates). This general definition is reusable for all sentences consisting of a verb and a subject.

3.2.4 Type-checking is Well-formedness Checking

This part shows how to use the general definition of the section 3.2.3. By instanciation of the generic model $frame\_verb0$, we define the representation of the sentence *A car is moving* as :

```
Definition a_car_is_moving_gen :=
 (frame_verb0 car_is_vehiculeAmoteur
              move is_car).
```

or, as well, the sentence *A Rolls Royce is moving* as :

```
Definition a_RollsRoyce_is_moving :=
 (frame_verb0 RollsRoyce_is_car
              move is_RollsRoyce).
```

The Coq system performs the type-checking of these instanciations and so, it establishes the sentence's well-formedness. Other generic models based on verb's use have been implemented in Coq. For example, we describe several categories of sentences which are composed of a verb, a subject and a complement. But this is yet a case study and we plan to develop others models of sentences.

4 Conclusion and Perspectives

The work presented in this paper aims at studying the capabilities of the Coq system, in the field of semantic representation and conceptual analysis for natural language processing. The relevance of our encoding has been motivated by the particular features of the Coq system. It provides an unifying framework with a rich type system that allows to straightforwardly specify the underlying conceptual tree as well as to analyse semantic representations. The paper makes a connection with the use of resources, namely French Wordnet (WOLF) and a French lexicon of verbs (LVF). The approach can be summarized as follows :

1. definition of formal models for representing sentences by taking advantage of the Coq type system and its particularly rich language (polymorphism, higher-order logic, coercion mechanism, module system),
2. implementation of general specifications of sentences that can be checked for a conceptual analysis based on types,
3. use of type-checking algorithms involved into the Coq system,
4. combination of our formal development with WOLF for describing the hierarchy of concepts (encoded in Coq with types) and with LVF for taking into account the verb's lexicon and their semantic features.

We plan to improve current results by automating the importation of WOLF and LVF into the Coq system in order to work on complete resources of french language.

References

Houda ANOUN. 2005. Reasoning on Multimodal Logic with the Calculus of Inductive Constructions. In *Logic for Programming Artificial Intelligence Reasoning*.

Coq Development Team, INRIA, 2014. *The Coq Proof Assistant. Reference manual, v8.4*.

Yann COSCOY. 2000. *Explication textuelles de preuves pour le calcul des constructions inductives*. Thèse d'université, Université de Nice-Sophia-Antipolis.

Philippe DE GROOTE and Christian RÉTORÉ. 1996. Semantic Readings of Proof Nets. In Oehrle D. Kruijff G., Morrill G., editor, *Formal Grammar*, pages 57–70. FoLLI.

Jean DUBOIS and Franoise DUBOIS-CHARLIER. 1997. *Les verbes français*. Larousse-Bordas.

Christiane ed. FELLBAUM. 1998. Wordnet : An Electronic Lexical Database. In *MIT Press*.

Valérie HANOKA and Benoît SAGOT. 2012. Wordnet Creation and Extension make Simple : A Multilingual Lexicon-based Approach using Wiki Resources. In *LREC*.

Alain LECOMTE. 1996. Grammaire et théorie de la preuve : une introduction. In *Traitement automatique des langues*, volume 37, pages 1–38.

Zhaohui LUO. 2010. Type-theoretical Semantics with Coercive Subtyping. In *SALT*.

Zhaohui LUO. 2012. Common Nouns as Types. In *LACL*.

Michael MOORTGAT and Richard MOOT. 2002. Using the Spoken Dutch Corpus for type-logical grammar induction. In *LREC*.

Michael MOORTGAT. 1996. Categorial Type Logics. In J.Van Benthem and A. Ter Meulen, editors, *Handbook of Logic and Language*, North-Holland Elsevier-Amsterdam, pages 93–177.

Richard MOOT and Christian RÉTORÉ. 2011. The Logic of Categorial Grammars. volume 6850. LNCS.

Reinhard MUSKENS. 2011. Type-logical Semantics. In E. Craig, editor, *Routledge Encyclopedia of Philosophy Online*. Routledge.

Aarne RANTA. 2009. GF: a Multilingual Grammar Formalism. In *Language and Linguistics Compass*, volume 3.

Christian RETORÉ. 2001. Systèmes déductifs et traitement des langues, un panorama des grammaires catégorielles. volume 20 of *TSI*, pages 301–336.